Control Engineering Practice 74 (2018) 168–176

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Improving low cost sensor based vehicle positioning with Machine Learning

Ikram Belhajem *, Yann Ben Maissa, Ahmed Tamtaoui

Department of Telecommunications, Networks and Service Systems, National Institute of Posts and Telecommunications, Allal El Fassi Avenue, Rabat, Morocco

ARTICLE INFO

Keywords:
Smart city
Intelligent transportation systems
Low cost
Machine Learning
Neural Networks
Support Vector Machines
Genetic Algorithm
Particle Swarm Optimization
Global Positioning System
Dead Reckoning

ABSTRACT

Current fleet management solutions rely on real time vehicle information to efficiently resolve transportation problems. In this study, a novel robust approach based on combining the Extended Kalman Filter (EKF) with Machine Learning techniques, Neural Networks or Support Vector Machines, is introduced to improve the accuracy of vehicle position estimation and circumvent the EKF limitations. The proposed solution guarantees also a low cost by using the Global Positioning System enhanced with Dead Reckoning integrated sensors. To verify our approach, extensive simulation tests are conducted on field data sets and very promising progress is obtained in the estimated vehicle position.

1. Introduction

Context. The main concern of a smart city is to improve the quality of life despite the challenging problems emerged by the rapid urban growth. It relies on the use of information and communication technologies in various functional domains to ensure an intelligent urban development and a sustainable socio-economic growth (Neirotti, Marco, Cagliano, Mangano, & Scorrano, 2014). Smart cities integrate, among other solutions, intelligent transportation systems (ITS) that offer a safe transport environment with regard to cost reduction and efficiency enhancement. One way to meet these expectations is the deployment of fleet management systems as they ensure a real time tracking and an online control monitoring for a fleet of vehicles (Gowda & Gopalakrishna, 2015). This has a positive impact on optimizing vehicle routing and reducing fuel consumption but requires having clear visibility on each vehicle location under real time and low cost constraints.

Most navigation systems rely on the Global Positioning System (GPS) since it determines the location, altitude and velocity based on satellite signals received. However, the GPS performance is limited by different factors such as the effect of atmospheric disturbances and multipath phenomenon, where one or more signals arrive at the GPS antenna by indirect paths (Bilich & Larson, 2007), that can happen within urban environment or beneath dense foliage. With the use of Differential GPS (DGPS), it is possible to remove some GPS errors through corrections that a reference GPS receiver at a known location provides. Another option is to combine GPS and Inertial Navigation Systems (INS) for measuring the position without interruption. The INS are self-contained systems that regroup a set of sensors: three or more accelerometers and gyroscopes and a navigation computer used to calculate the position, velocity and attitude.

To fuse data coming from multiple sensors, a Kalman Filter (KF) is typically applied. The filter is a recursive algorithm that optimally estimates an unknown state of a linear dynamic system starting from uncertain and noisy observations (Gao & Harris, 2002). For non-linear systems, the Extended KF (EKF) is adopted through a linearization procedure using Taylor series expansion.

Problem. The impact of multipath errors on positioning estimates remains even when using differential *high cost* corrections. Additionally, the INS impose restrictions on the environments where they are implemented because of their computational *complexity*. The KF performance depends on the accuracy of the stochastic modeling of sensors. It also requires a priori knowledge of system noises and measurement errors. Relatively, the KF estimated position quickly *diverges* when GPS outage happens.

Contribution. This paper explores the application of a novel low cost robust approach combining EKF and Machine Learning (ML) techniques notably the Neural Networks (NN) or Support Vector Machines (SVM) for an optimal real time car positioning in a smart city. For this, the GPS aided Dead Reckoning (DR) sensors (i.e., an odometer to provide the distance traveled by the vehicle and a gyrometer to measure the angular velocity) is used. It is a simple and easy to deploy mechanism. During

* Corresponding author.
E-mail address: belhajem@inpt.ac.ma (I. Belhajem).

https://doi.org/10.1016/j.conengprac.2018.03.006
Received 14 April 2017; Received in revised form 16 January 2018; Accepted 4 March 2018
0967-0661/

List of acronyms

BPLA	Backpropagation Learning Algorithm
DGPS	Differential GPS
DR	Dead Reckoning
EKF	Extended KF
GA	Genetic Algorithm
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation Systems
ITS	Intelligent transportation systems
KF	Kalman Filter
LS-SVM	Least squares SVM
MAE	Mean absolute error
MEMS	Micro-Electro-Mechanical Systems
MFNN	Multilayer feed-forward NN
ML	Machine Learning
MSE	Mean squared error
NN	Neural Networks
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
RMSE	Root mean square error
SRM	Structural Risk Minimization
SVM	Support Vector Machines
UKF	Unscented KF
ε-SVR	ε-Support Vector Regression

GPS signal presence, the EKF computes the vehicle position and the ML module is trained in the meantime with various optimization techniques (i.e., Backpropagation Learning Algorithm (BPLA), grid search, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)) to model the position errors. When GPS outage occurs, the ML module limits the EKF drawbacks by compensating the additional position errors.

Contents. The rest of this paper is organized into several sections. Section 2 presents an outline of some relevant works on this topic; Section 3 covers the essential background on EKF and ML techniques; Section 4 describes our suggested approach and Section 5 details the simulation results.

2. Related works

Several works (Lucet et al., 2009; Shen, Georgy, Korenberg, & Noureldin, 2011) suggest that vehicle positioning can be achieved by combining GPS and DR system due to their complementarity. The DR system relies on odometer and gyrometer readings to estimate the position given known initial values. Although its full availability and ability to operate independently of any signal outage, the DR precision is limited in a long term because of time growing errors such as bias drift and scale factor change. Consequently, the integrated GPS/DR system helps to mitigate the DR errors when GPS is available and to overcome the positioning discontinuity induced by GPS signal outages. Furthermore, such system costs much lower than other solutions as presented in Table 1.

Godha and Cannon (2005) suggest using the KF to integrate data coming from the system DGPS/INS based on Micro-Electro-Mechanical Systems (MEMS) for the estimation of sensor errors. The MEMS technology is characterized mainly by its affordable cost and small size but suffers from a rapid growth of errors in the position estimation when operating for a long time in stand-alone mode (Park & Gao, 2008). In their work, St-Pierre and Gingras (2004) use a GPS receiver coupled to IMU/odometer/inclinometer sensors for position estimation of land vehicle navigation applications. In this sense, they compare the performance of Unscented KF (UKF) to that of EKF. Experiments show the inability of UKF to provide better results during GPS signal outages in addition to its high computational time.

As an alternative to KF, Chiang et al. (2003) propose the use of multilayer feed-forward NN (MFNN) with BPLA for DGPS/INS data fusion. The principle of NN is processing information in parallel so as to map inputs to desired outputs by learning from given samples. NN are commonly trained using BPLA but there are other techniques namely GA and PSO (Malleswaran, Vaidehib, & Sivasankari, 2014). However, the solution DGPS/INS is very expensive as depicted in Table 1. Also the ML models require that observations should be sufficient enough to enhance the precision of estimated position.

Nevertheless, it can be inadequate to substitute the EKF with NN for all case studies: Belhajem, Ben Maissa, and Tamtaoui (2016) introduce a hybrid approach combining EKF and NN based on a low cost GPS/DR system, the goal is to deal with the precision degradation during the EKF prediction phase. Two MFNN are trained to learn the north and east position errors for given inputs that include time elapsed since last GPS measurement, velocity and heading angle. They can then compensate the EKF drifts when no GPS signal is available. In the same perspective, Xu, Li, Rizos, and Xu (2010) describe the GPS/INS integration module relying on least squares SVM (LS-SVM) and KF hybrid method to correct the INS errors when losing the GPS signal. They obtained better results than INS-only solution during GPS signal outages. SVM are supervised learning algorithms whose objective is to build a model well suited for unseen samples, the common way to select optimally the SVM parameters is the grid search but it is possible to use GA (Tan, Wang, Jin, & Meng, 2015) or PSO (Lin, Ying, Chen, & Lee, 2008).

3. Background

In this section, an overview of the EKF, NN and SVM is presented. Also, a background of evolutionary algorithms notably the GA and PSO is covered.

Table 1
Approximate cost of positioning solutions.

Study	Proposed solution	Experimental setup	Approximate cost
Our approach	GPS/DR system	• GPS Trimble AG132 antenna • Built-in anti-lock braking system • Grove Gyro	<200 $
Bonnabel, Deschaud, and Salaün (2011)	GPS/Inertial Measurement Unit (IMU) coupled to two wheel speed sensors	• GPS Trimble AG132 antenna • Crossbow VG600 IMU • Two speedometers	>11,100 $
Chiang, Noureldin, and El-Sheimy (2003)	DGPS/INS	• NovAtel BDS GPS/IMU (the IMU is a Honeywell HG1700)	>34,000 $
Bhatt, Aggarwal, Devabhaktuni, and Bhattacharya (2012)	GPS/INS	• NovAtel OEM GPS • Crossbow IMU 300CC-100	>10,500 $

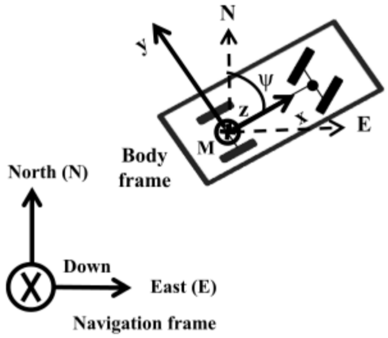

Fig. 1. Vehicle kinematic model.

3.1. Extended Kalman filter

The EKF is a recursive algorithm that tries to minimize error covariance by repeating the prediction-update process until the last observation (Gao & Harris, 2002). The filter uses a linearized process model to predict an a priori state of the system, then a correction is made taking in consideration any sensor measurements.

Modeling

Fig. 1 presents the kinematic model of a front-wheel drive vehicle. For vehicle dynamics analysis, any coordinates defined in the body frame (i.e., the origin M is located midway the rear axle and the x-axis is aligned with the vehicle longitudinal axis) must be converted in the north-east-down navigation frame. The kinematic equations (Lucet et al., 2009) mentioned below describe the vehicle position transformed into a navigation frame where (N_{k+1}, E_{k+1}) are north and east values recorded at the $k+1$ epoch, ψ_{k+1} represents the heading while ds_{k+1} and $d\psi_{k+1}$ are the distance traveled and heading variation between k and $k+1$, respectively. For the GPS antenna, it is separated from the rear axle by distances D_x and D_y.

$$\begin{cases} N_{k+1} = N_k + ds_{k+1} \cdot sinc(\frac{d\psi_{k+1}}{2}) \cdot cos(\psi_k + \frac{d\psi_{k+1}}{2}) \\ \quad - d\psi_{k+1} \cdot (D_x \cdot sin(\psi_k) + D_y \cdot cos(\psi_k)) \\ E_{k+1} = E_k + ds_{k+1} \cdot sinc(\frac{d\psi_{k+1}}{2}) \cdot sin(\psi_k + \frac{d\psi_{k+1}}{2}) \\ \quad + d\psi_{k+1} \cdot (D_x \cdot cos(\psi_k) - D_y \cdot sin(\psi_k)) \\ \psi_{k+1} = \psi_k + d\psi_{k+1}. \end{cases} \tag{1}$$

Prediction phase

The kinematic model presented in Eq. (1) can be written in a more compact form considering a Gaussian white noise w_k:

$$X_{k+1} = f(X_k, u_k) + w_k \tag{2}$$

where $X_{k+1} = (N_{k+1}, E_{k+1}, \psi_{k+1})^T$ is the state vector at time epoch $k+1$, $f(\cdot)$ is a nonlinear function representing the state evolution and u_k is the control vector of distance traveled and heading variation. As a first step, the filter initializes the state and the error covariance that must be known beforehand. Thereafter, the prediction mode starts by projecting the state and its associated error covariance:

$$\hat{X}^-_{k+1} = f(\hat{X}_k, u_k) \tag{3}$$

$$P^-_{k+1} = F_{k,x} P_k F^T_{k,x} + F_{k,u} Q^u_k F^T_{k,u} + Q^{mod}_k \tag{4}$$

where $\hat{X}^-_{k+1}$ is the a priori estimate of the state at time epoch $k+1$, P^-_{k+1} is the a priori estimate error covariance, $F_{k,x}$ and $F_{k,u}$ are Jacobian matrices with respect to the state and the control vectors respectively, Q^u_k represents the covariance matrix associated with the noise in the DR sensors while Q^{mod}_k is the covariance matrix of the kinematic model mismatch.

Correction phase

The correction phase starts simultaneously with the arrival of new GPS measurements at time epoch $k+1$. The measurement vector is $Z_{k+1} = (N_{GPS_{k+1}}, E_{GPS_{k+1}})^T$ presented as:

$$Z_{k+1} = h(X_{k+1}) + v_k \tag{5}$$

where $h(\cdot)$ is a nonlinear function that relates the state to the measurement and v_k is assumed to be a Gaussian white noise. At this stage, the Kalman gain K_{k+1}, a posteriori estimate of the state $\hat{X}^+_{k+1}$ and a posteriori estimate error covariance P^+_{k+1} are computed:

$$K_{k+1} = P^-_{k+1} H^T (H_{k+1} P^-_{k+1} H^T + R_{k+1})^{-1} \quad \text{with} \quad \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{6}$$

$$\hat{X}^+_{k+1} = \hat{X}^-_{k+1} + K_{k+1}(Z_{k+1} - H\hat{X}^-_{k+1}) \tag{7}$$

$$P^+_{k+1} = (I - K_{k+1} H) P^-_{k+1} \tag{8}$$

where R_{k+1} denotes the covariance matrix of the measurement and I is the identity matrix.

3.2. Neural networks

NN are parallel distributed processors which imitate the brain structure in such a way that they are composed of many neurons linked by synaptic weights and organized in layers (Alpaydin, 2010). The commonly used network architecture is MFNN that consist of an input layer, one or more hidden layers and an output layer. The MFNN are trained according to learning rules that control the weight adjustment.

Usually, the BPLA is applied for weight update by computing the gradient descent so as to reduce the error function between the networks outputs and the desired values. The gradient descent, though, suffers from slow convergence and hence comes up the idea of using the momentum that considers previous weight update in the current change as follows (Alpaydin, 2010):

$$w^t_{ij} = w^{t-1}_{ij} - \eta \frac{\partial E^t}{\partial w^t_{ij}} + \alpha \Delta w^{t-1}_{ij} \tag{9}$$

where w^t_{ij} is the weight connecting neuron j to neuron i at time epoch t, η is the learning rate, $\frac{\partial E^t}{\partial w^t_{ij}}$ is the gradient of the error function E^t with respect to the weight w^t_{ij}, α is the momentum term and Δw^{t-1}_{ij} denotes the previous weight change at time epoch $t-1$.

3.3. Support vector machines

SVM are ML algorithms that perform a non-linear mapping from the input space into a high dimensional feature space through the use of kernels (Cortes & Vapnik, 1995). For this, the Structural Risk Minimization (SRM) inductive principle is adopted to select a model with the smallest upper bound of the generalization error. SVM can be applied successfully to both classification (Lin et al., 2008) and regression (Tan et al., 2015; Xu et al., 2010) problems due to their generalization ability and capability to find global minima for any data (Basak, Pal, & Patranabis, 2007).

In our case, the ε-Support Vector Regression (ε-SVR) is chosen to infer the input–output functional relationship for training data. Based on a given set of training data points whose size is l $\{(x_i, y_i)\}$ $(i = 1, \dots, l)$, the goal of ε-SVR is to find a function $f(x)$ that has at most ε deviation from the targets y_i for all the training data and at the same time is as flat as possible (Basak et al., 2007). In addition, the trade-off between the flatness of f and the amount up to which deviations larger than ε are tolerated is controlled by changing the cost parameter C. The way ε-SVR solves the approximation problem is presented as:

$$f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha^*_i) K(x_i, x) + b \tag{10}$$

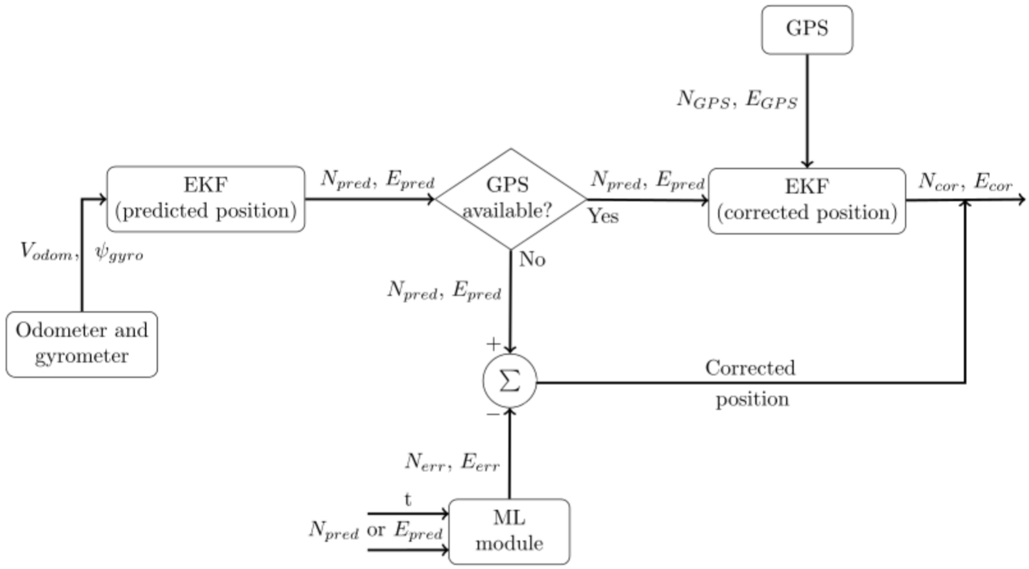

Fig. 2. Combination of EKF and ML techniques.

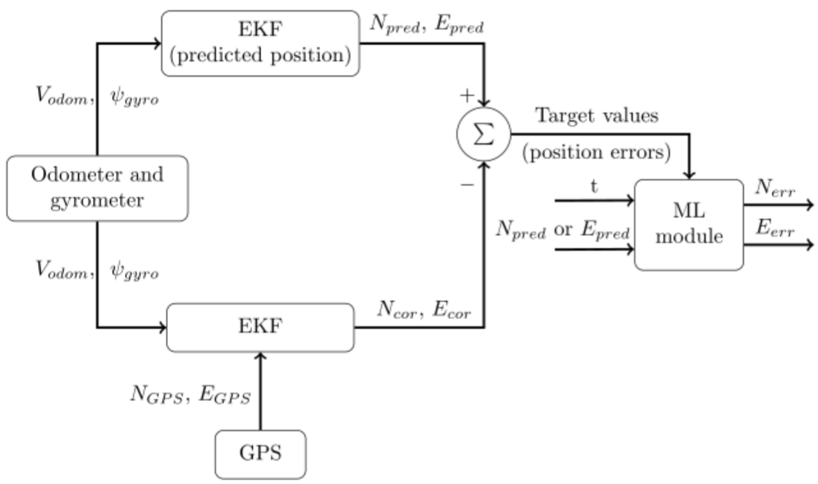

Fig. 3. Training phase of ML techniques.

where α_i and α_i^* are Lagrange multipliers, b is the bias and K is the kernel function equal to the inner product of two vectors x_i and x_j in the feature space $\phi(x_i)$ and $\phi(x_j)$. Here, the Radial Basis Function (RBF) kernel of parameter γ described in Eq. (11) is selected because of its tendency to yield good performance under general smoothness assumptions. The selection of ε-SVR parameters (i.e., ε, C and γ) is generally achieved by an exhaustive grid search over candidate parameter values in order to choose the optimal ones. Further details concerning the ε-SVR are contained in (Basak et al., 2007).

$$K_{RBF}(x_i, x_j) = \exp^{-\gamma\|x_i - x_j\|^2}. \tag{11}$$

3.4. Evolutionary algorithms

Here, the process of evolutionary algorithms namely GA and PSO is detailed.

Genetic algorithm

GA is a stochastic search and optimization method that explores a wide range of potential solutions to complex problems, based on the simulation of natural selection process (Holland, 1992). To this end, GA addresses the problem by setting a population of candidate solutions called chromosomes and each chromosome consists of a number of variables referred to as genes (Sastry, Goldberg, & Kendall, 2014).

The simplicity and flexibility of GA justify its use for parameter optimization in the training phase of NN (Malleswaran et al., 2014) or SVM (Tan et al., 2015). Firstly, an initial population of potential solutions is generated randomly across the search space. Then, each individual is evaluated by a given fitness function used to guide the evolution of good solutions. After that, individuals with higher fitness values are selected to reproduce. Once this done, two randomly chosen parent chromosomes are recombined with a certain probability p_c, known as the crossover probability, to create new offspring. Afterwards, the offspring chromosomes are subjected to mutation according to a defined probability p_m called the mutation probability. Finally, a replacement operator is applied to incorporate the newly created offspring solutions into the original parental population. These steps, except the initialization, are repeated until a terminating condition is satisfied (Sastry et al., 2014).

Particle swarm optimization

PSO is an evolutionary computation technique which highly depends on stochastic processes to optimize different types of problems (Kennedy & Eberhart, 1995). Inspired by the social behavior of animals, PSO is a population-based algorithm that represents each problem to solve by a swarm of candidate solutions called particles which fly in the search space looking for the optimal solution (Shi, 2004). Since the PSO is simple to implement and computationally efficient (Kennedy & Eberhart, 1995), it is used to optimize the NN weights (Malleswaran et al., 2014) or the SVM parameters (Lin et al., 2008) in the training stage.

The PSO algorithm begins by initializing a swarm of particles with random positions and velocities in the problem space. Each particle i is characterized by a position vector x_i, a velocity vector v_i and a personal best position p_i at which the best fitness value $pbest$ is achieved so far. Furthermore, the global best position p_g represents the position yielding the best fitness value $gbest$ obtained among all the $pbest$. At each iteration, every particle is evaluated by a chosen objective function. If the current value is better than $pbest$, $pbest$ is set equal to this value and p_i should be replaced by the current position. Relatively, the current $pbest$ of each particle is compared to $gbest$ and if it is better, $gbest$ takes this value and p_g is set equal to the current location. After that, the velocity and position of each particle are updated at iteration $k+1$ according to the following equations:

$$v_i(k+1) = wv_i(k) + c_1 r_1(k)(p_i(k) - x_i(k)) + c_2 r_2(k)(p_g(k) - x_i(k)) \tag{12}$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \tag{13}$$

where c_1 and c_2 are positive constants, $r_1(k)$ and $r_2(k)$ are random numbers uniformly distributed in the range [0,1] and w is the inertia weight employed to balance between global and local exploration abilities of the swarm. The PSO update process is repeated until a stopping criterion is met, usually reaching a target fitness value or a preset number of iterations (Shi, 2004).

4. Proposed approach for GPS/DR integration

The EKF efficiency depends on correct modeling of DR sensors and environment changes that influence the GPS quality. This is why the filter performance degrades and may diverge when no GPS signal exists. The ML techniques, NN or SVM, are then used in order to estimate the time-correlated errors of EKF predicted position during GPS signal outages.

Formally, the EKF starts with initializing the vehicle position and thereafter it computes a predicted position (N_{pred}, E_{pred}) when measurements of velocity V_{odom} and heading angle ψ_{gyro} arrive. Then, the filter corrects this estimated position (N_{cor}, E_{cor}) once new GPS observations are received. In periods of GPS signal outages, the ML module (i.e., NN or SVM) compensates the accumulated EKF errors as shown in Fig. 2.

Regarding the topology, the north (resp. east) ML module inputs are north (resp. east) EKF predicted position and time duration since last GPS measurement while the output is north (resp. east) position drift which is compared to the target position error. For the desired values, they are computed as a difference between positions that two parallel EKFs provide (cf. Fig. 3): one filter gives a corrected position while the other estimates a predicted one by removing intentionally the GPS signals (Goodall, Syed, & El-Sheimy, 2006).

During the training stage (illustrated in Fig. 3), the ML module is trained on various samples at the GPS frequency. Once the training on different examples is completed, the prediction stage starts and the ML module predicts the north and east position errors based on collected input data. Hence, the EKF predictions are compensated by the ML estimates to form the corrected vehicle position as presented in Fig. 2.

To approve the robustness of our approach, it is compared to existing solutions based only on ML techniques in particular NN (Chiang et al., 2003) or SVM (Xu et al., 2010). For this, two three-layer MFNN or two SVM are trained on different examples so as to predict efficiently the vehicle position components during GPS signal outages. Concerning their topology, the north (resp. east) NN or SVM inputs consist of velocity $V_{odom}(t)$ and heading angle $\psi_{gyro}(t)$ while the output is north (resp. east) position to be matched by the desired north (resp. east) GPS position ($N_{GPS}(t), E_{GPS}(t)$).

Table 2
Implementation parameters of BPLA, GA and PSO.

Learning Algorithm	Parameter	Value
BPLA	Number of input neurons	2
	Number of hidden neurons	5
	Number of output neurons	1
	Learning rate	0.1
	Momentum term	0.9
GA	Population size	20
	Selection	Tournament
	Crossover	Single-point
	Mutation	Random
	Crossover probability p_c	0.6
	Mutation probability p_m	0.2
PSO	Swarm size	20
	Inertia weight w	0.729
	Positive constant c_1	1.49445
	Positive constant c_2	1.49445
	Numbers r_1 and r_2	Random

5. Simulation results

In what follows, extensive simulation tests on the Institut Pascal Data Sets (Korrapati, Courbon, Alizon, & Marmoiton, 2013) are carried out in order to examine the effectiveness of our approach.

5.1. Test vehicle prototype

The Institut Pascal Data Sets (Korrapati et al., 2013) were collected via the VIPALAB platform that embeds multiple proprioceptive and exteroceptive sensors. Practically, our solution is tested with three data sets of a GPS receiver (i.e., uBlox-6T at the sampling rate of 1 Hz), an odometer and a MLX90609-N2 gyrometer at 50 Hz. The road trajectory is CEZEAUX-Heko (given in Fig. 4) acquired across the CEZEAUX campus over a distance of 4.2 km, this sequence is selected because it contains challenging stretches of curves and straight lines.

5.2. Results and discussion

The validity of our adopted approach is examined by dividing the field data into training and prediction parts. The first one lasts 19 min, in which seven GPS outages (shown in Fig. 4) are simulated to train the ML module. The duration of each outage is 60 s to leave the EKF position errors enough time to grow. The last part is dedicated to the testing phase where ML techniques predict the position errors.

The NN or SVM are trained using a batch-incremental approach: instances of input and target data are presented for every outage so as to learn the NN or SVM parameters. The aim of the training is to get a small mean squared error (MSE) between the NN or SVM outputs and the desired values (see Fig. 3). To check the ML module generalization capacity towards unseen data, a hold-out validation is introduced due to the fact that it is a simple and not time-consuming method. The idea behind is to split randomly the data into two sets, one is for training while the remaining is for validating the trained model (Arlot & Celisse, 2010). The NN topology selected empirically, the implementation parameters and the training time of BPLA, grid search, GA and PSO are given in Tables 2 and 3.

The training phase of ML techniques is achieved correctly since the MSE reached is less than 5×10^{-5} m^2. As a consequence, NN or SVM trained with different optimization techniques (i.e., BPLA, grid search, GA and PSO) have very similar outputs as the target values. The NN and SVM training results are presented in Figs. 5 and 6, it is important to note that time between training outages does not appear in these plots.

In the prediction phase, the GPS signal outages were simulated for durations of 90 s and 60 s to test our hybrid approach. In this case, the NN or SVM generate estimates of position errors to correct the EKF predictions, with respect to dynamic inputs and latest NN or SVM

Table 3
Training time of BPLA, Grid search, GA and PSO.

Outages	NN			SVM		
	BPLA	GA	PSO	Grid search	GA	PSO
North position						
Outage 1	19 s	23 s	13 s	8 s	1 s	6 s
Outage 2	19 s	22 s	13 s	7 s	1 s	6 s
Outage 3	19 s	23 s	42 s	8 s	13 s	7 s
Outage 4	19 s	22 s	6 s	7 s	3 s	2 s
Outage 5	18 s	22 s	6 s	7 s	8 s	1 s
Outage 6	18 s	22 s	3 s	7 s	2 s	6 s
Outage 7	18 s	23 s	4 s	7 s	4 s	6 s
East position						
Outage 1	19 s	22 s	20 s	8 s	5 s	6 s
Outage 2	19 s	23 s	3 s	7 s	1 s	1 s
Outage 3	19 s	22 s	5 s	7 s	6 s	6 s
Outage 4	19 s	23 s	14 s	7 s	2 s	1 s
Outage 5	19 s	23 s	11 s	7 s	34 s	6 s
Outage 6	19 s	23 s	3 s	7 s	2 s	6 s
Outage 7	19 s	22 s	4 s	8 s	3 s	1 s

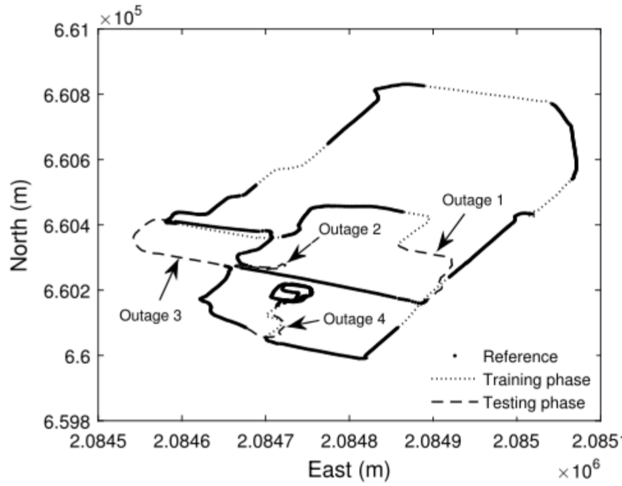

Fig. 4. Field test trajectory.

optimized parameters. Figs. 7 and 8 present the results of GPS test outages (shown in Fig. 4) for NN and SVM.

To analyze these graphs, two different evaluation indicators are calculated for each outage: the root mean square error (RMSE) and the mean absolute error (MAE). They are measured by the formulas below:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_{ref,i} - Y_{pred,i})^2} \quad (14)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y_{ref,i} - Y_{pred,i}\right| \quad (15)$$

where $Y_{ref,i}$ is the reference EKF updated position by GPS measurements at time epoch i, $Y_{pred,i}$ is the predicted position by means of EKF or our suggested approach while N is the total number of predictions. Tables 4 and 5 list the testing results for north and east position estimates. The hybridization of EKF and ML techniques generates good performance results when compared to the EKF. A major drop in RMSE and MAE values leads to a promising improvement in each GPS outage even for periods of 90 s. In addition, the training time of SVM is generally faster than NN since only three parameters (i.e., ε, C and γ) must be optimized. It appears also that PSO needs less time to converge than other optimization algorithms (as presented in Table 3). As a consequence, the EKF/SVM trained with PSO is more appropriate for

Table 4
Improvement of position estimate using EKF/NN hybrid approach during GPS test outages.

Outages	EKF		EKF/NN (BPLA)		Improvement (%)		EKF/NN (GA)		Improvement (%)		EKF/NN (PSO)		Improvement (%)	
	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)
North position														
Outage 1 (90 s)	732.47	630.57	**214.11**	**184.74**	**70**	**70**	**213.22**	**182.85**	**70**	**71**	**217.95**	**187.43**	**70**	**70**
Outage 2 (60 s)	408.82	343.18	**103.54**	**91.70**	**74**	**73**	**100.87**	**87.28**	**75**	**74**	**105.91**	**94.28**	**74**	**72**
Outage 3 (90 s)	715.54	613.58	**207.08**	**176.75**	**71**	**71**	**209.42**	**178.29**	**70**	**70**	**198.72**	**169.07**	**72**	**72**
Outage 4 (60 s)	418.99	372.16	**97.45**	**74.97**	**76**	**79**	**95.55**	**71.38**	**77**	**80**	**104.97**	**80.45**	**74**	**78**
East position														
Outage 1 (90 s)	640.44	545.17	**227.14**	**191.52**	**64**	**64**	**232.36**	**197.31**	**63**	**63**	**240.67**	**204.32**	**62**	**62**
Outage 2 (60 s)	419.84	359.72	**29.00**	**25.23**	**93**	**92**	**28.24**	**25.67**	**93**	**92**	**30.08**	**27.65**	**92**	**92**
Outage 3 (90 s)	681.06	606.22	**269.65**	**248.61**	**60**	**58**	**276.86**	**255.78**	**59**	**57**	**284.51**	**262.46**	**58**	**56**
Outage 4 (60 s)	377.17	319.96	**8.14**	**6.27**	**97**	**98**	**8.86**	**7.69**	**97**	**97**	**14.34**	**11.69**	**96**	**96**

Table 5
Improvement of position estimate using EKF/SVM hybrid approach during GPS test outages.

Outages	EKF		EKF/SVM (Grid search)		Improvement (%)		EKF/SVM (GA)		Improvement (%)		EKF/SVM (PSO)		Improvement (%)	
	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)
North position														
Outage 1 (90 s)	732.47	630.57	**212.65**	**182.73**	**70**	**71**	**211.87**	**180.48**	**71**	**71**	**212.21**	**182.18**	**71**	**71**
Outage 2 (60 s)	408.82	343.18	**103.79**	**91.15**	**74**	**73**	**104.79**	**93.84**	**74**	**72**	**104.52**	**91.99**	**74**	**73**
Outage 3 (90 s)	715.54	613.58	**207.15**	**176.08**	**71**	**71**	**206.50**	**173.73**	**71**	**71**	**206.65**	**175.66**	**71**	**71**
Outage 4 (60 s)	418.99	372.16	**97.49**	**72.78**	**76**	**80**	**97.26**	**72.65**	**76**	**80**	**98.12**	**73.51**	**76**	**80**
East position														
Outage 1 (90 s)	640.44	545.17	**235.36**	**201.19**	**63**	**63**	**236.71**	**202.89**	**63**	**62**	**237.23**	**203.58**	**62**	**62**
Outage 2 (60 s)	419.84	359.72	**28.76**	**27.11**	**93**	**92**	**27.69**	**25.61**	**93**	**92**	**28.26**	**26.39**	**93**	**92**
Outage 3 (90 s)	681.06	606.22	**280.67**	**259.62**	**58**	**57**	**282.98**	**262.33**	**58**	**56**	**283.63**	**263.01**	**58**	**56**
Outage 4 (60 s)	377.17	319.96	**11.20**	**9.67**	**97**	**96**	**14.52**	**12.83**	**96**	**95**	**13.94**	**12.34**	**96**	**96**

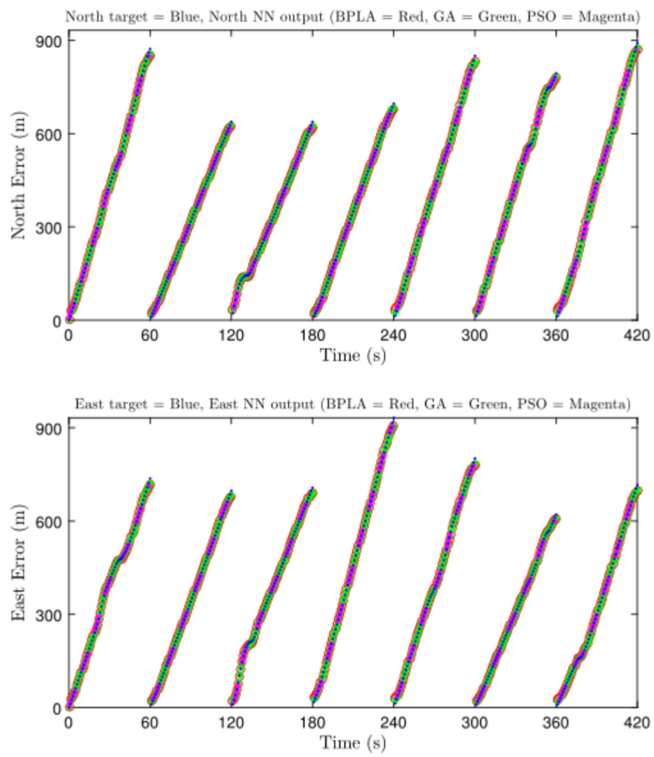

Fig. 5. North and east NN training results.

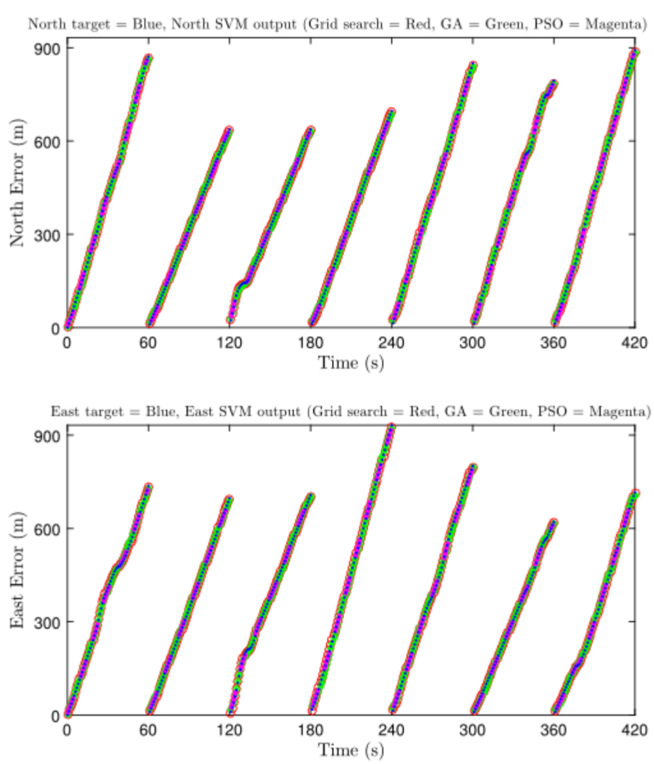

Fig. 6. North and east SVM training results.

positioning applications due to the enhancement of position accuracy and the short training time.

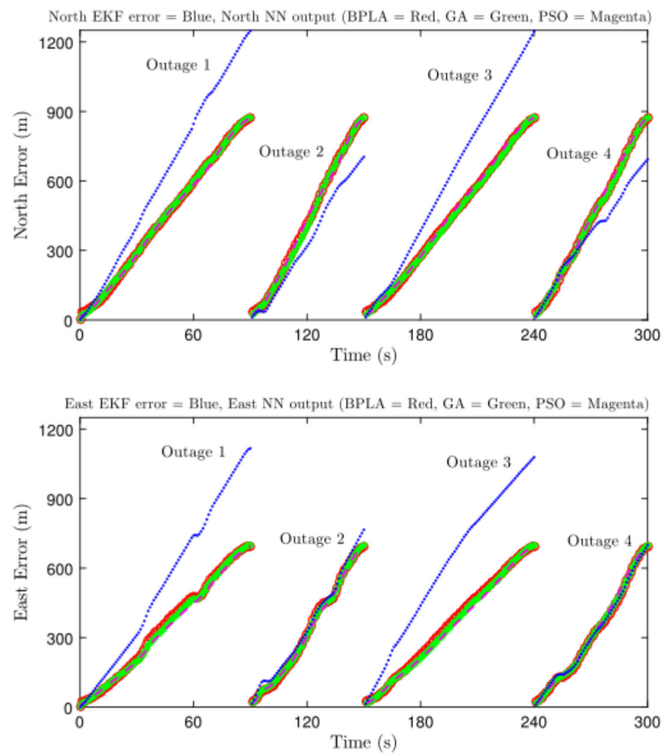

Fig. 7. North and east NN testing results.

Similarly, the solutions relying just on ML techniques are trained on different batches of inputs/targets with a hold-out validation. The parameter optimization of north and east NN or SVM is conducted by BPLA and grid search, respectively. The training stage is performed in such a way that the MSE is less than 5×10^{-2} m^2 given constraints of real time applications and results indicate that the outputs are close enough to the desired values. Table 6 presents the time required for training the ML algorithms.

During GPS test outages, NN or SVM give a good yield according to RMSE and MAE statistics. Hence, they provide a better overall improvement (i.e., calculated as an average of all individual outage improvements) over the EKF predictions as depicted in Table 7. However, this performance is still lower than our approach EKF/SVM trained with PSO whose overall improvement varies between 73% and 77%.

6. Conclusion

This paper introduces a *reliable* approach to ensure an uninterrupted estimation of vehicle position for fleet management systems in a smart city. The originality of this study stems from using EKF and ML techniques, NN or SVM, to fuse data coming from GPS augmented with *low cost* DR system. The EKF estimates the position during GPS signal presence while the ML module is trained in parallel with different optimization techniques (i.e., BPLA, grid search, GA, PSO). When no GPS information is available, the ML module is able to predict the position drifts and compensate it from the EKF erroneous position estimates. Simulation results imply the effectiveness of our hybrid approach to improve the positioning exactitude compared to EKF, NN and SVM. In particular, the EKF/SVM trained with PSO combines good precision with less training time which makes it more suitable for positioning solutions.

Future work. The validation of our proposed approach was done through the use of simulated GPS outages. Unfortunately, GPS quality degrades in real cases and to take this factor into consideration, our vehicle Arduino based prototype is currently under implementation.

Table 6
Training time of BPLA and Grid search.

Outages	NN (BPLA)		SVM (Grid search)	
	North	East	North	East
Outage 1	20 s	19 s	10 s	10 s
Outage 2	20 s	19 s	10s	10 s
Outage 3	19 s	19 s	10s	10 s
Outage 4	20 s	19 s	11s	11 s
Outage 5	19 s	20 s	10s	10 s
Outage 6	19 s	19 s	10s	10 s
Outage 7	19 s	20 s	10s	10 s

Table 7
Improvement of position estimate using NN or SVM during GPS test outages.

Outages	NN		SVM	
	RMSE (m)	MAE (m)	RMSE (m)	MAE (m)
North position				
Outage 1 (90 s)	165.49	156.64	155.03	145.77
Outage 2 (60 s)	131.73	127.77	144.02	139.57
Outage 3 (90 s)	104.39	92.68	90.59	70.93
Outage 4 (60 s)	315.81	314.04	305.99	304.84
Overall improvement (%)	**63**	**59**	**63**	**60**
East position				
Outage 1 (90 s)	30.65	26.15	23.27	20.23
Outage 2 (60 s)	180.93	178.31	180.99	178.36
Outage 3 (90 s)	293.92	291.93	304.86	303.68
Outage 4 (60 s)	166.46	164.84	172.59	171.12
Overall improvement (%)	**65**	**61**	**65**	**60**

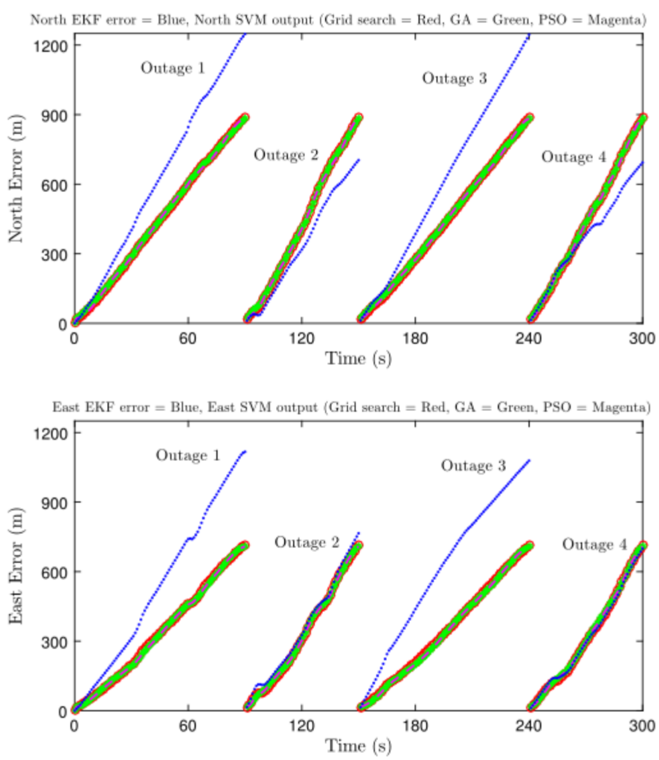

Fig. 8. North and east SVM testing results.

Acknowledgment

This work was supported by the National Center of Scientific and Technical Research (CNRST), Morocco (3INPT2015).

References

Alpaydin, E. (2010). Introduction to machine learning. Chapter: Multilayer perceptrons (pp. 233–277) http://dx.doi.org/10.1017/S0269888910000056.

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, *4*, 40–79. http://dx.doi.org/10.1214/09-SS054.

Basak, D., Pal, S., & Patranabis, D. C. (2007). Support vector regression. *Neural Information Processing — Letters and Reviews*, *11*(10), 203–224.

Belhajem, I., Ben Maissa, I., & Tamtaoui, A. (2016). A hybrid low cost approach using extended Kalman filter and neural networks for real time positioning. In *2016 international conference on information technology for organizations development, IT4OD* (pp. 1–5). http://dx.doi.org/10.1109/IT4OD.2016.7479298.

Bhatt, D., Aggarwal, P., Devabhaktuni, V., & Bhattacharya, P. (2012). A new source difference artificial neural network for enhanced positioning accuracy. *Measurement Science & Technology*, *23*(10), 1–11. http://dx.doi.org/10.1088/0957-0233/23/10/105101.

Bilich, A., & Larson, K. M. (2007). Mapping the GPS multipath environment using the signal-to-noise ratio (SNR). *Radio Science*, *42*(6), 1–16. http://dx.doi.org/10.1029/2007RS003652.

Bonnabel, S., Deschaud, J. E., & Salaün, E. (2011). A simple nonlinear filter for low-cost ground vehicle localization system. In *Proceedings of the 50th IEEE conference on decision and control and european control conference* (pp. 3270–3275). http://dx.doi.org/10.1109/CDC.2011.6161262.

Chiang, K. W., Noureldin, A., & El-Sheimy, N. (2003). Multisensor integration using neuron computing for land-vehicle navigation. *GPS Solutions*, *6*(4), 209–218. http://dx.doi.org/10.1007/s10291-002-0024-4.

Cortes, C., & Vapnik, V. (1995). Support-vector Networks. *Machine Learning*, *20*(3), 273–297. http://dx.doi.org/10.1023/A:1022627411411.

Gao, J., & Harris, C. (2002). Some remarks on kalman filters for the multisensor fusion. *Information Fusion*, *3*(3), 191–201. http://dx.doi.org/10.1016/S1566-2535(02)00070-2.

Godha, S., & Cannon, M. E. (2005). Integration of DGPS with a low cost MEMS-based Inertial Measurement Unit (IMU) for land vehicle navigation application. In *ION GPS*, vol. 5 (pp. 1–13).

Goodall, C., Syed, Z., & El-Sheimy, N. (2006). Improving INS/GPS navigation accuracy through compensation of Kalman filter errors. In *IEEE vehicular technology conference* (pp. 1–5). http://dx.doi.org/10.1109/VTCF.2006.578.

Gowda, V. R. C., & Gopalakrishna, K. (2015). Real time vehicle fleet management and security system. In *2015 IEEE recent advances in intelligent computational systems, RAICS* (pp. 417–421). http://dx.doi.org/10.1109/RAICS.2015.7488452.

Holland, J. H. (1992). Genetic algorithms Computer programs that "evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand. *Scientific American*, *267*(1), 66–72. http://dx.doi.org/10.1038/scientificamerican0792-66.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *IEEE international conference on neural networks, 1995. Proceedings, vol. 4* (pp. 1942–1948).

Korrapati, H., Courbon, J., Alizon, S., & Marmoiton, F. (2013). "The Institut Pascal Data Sets": un jeu de données en extérieur, multicapteurs et datées avec réalité terrain, données d'étalonnage et outils logiciels. In *ORASIS 2013*.

Lin, S., Ying, K. C., Chen, S. C., & Lee, Z. J. (2008). Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications*, *35*(4), 1817–1824. http://dx.doi.org/10.1016/j.eswa.2007.08.088.

Lucet, E., Betaille, D., Donnay Fleury, N., Ortiz, M., Salle, D., & Canou, J. (2009). Real-time 2D localization of a car-like mobile robot using dead reckoning and GPS, with satellite masking prediction. In *Accurate localization for land transportation workshop* (pp. 15–18).

Malleswaran, M., Vaidehib, V., & Sivasankari, N. (2014). A novel approach to the integration of GPS and INS using recurrent neural networks with evolutionary optimization techniques. *Aerospace Science and Technology*, *32*(1), 169–179. http://dx.doi.org/10.1016/j.ast.2013.09.011.

Neirotti, P., Marco, A., Cagliano, A., Mangano, G., & Scorrano, F. (2014). Current trends in smart city initiatives: Some stylised facts. *Cities*, *38*, 25–36. http://dx.doi.org/10.1016/j.cities.2013.12.010.

Park, M., & Gao, Y. (2008). Error and performance analysis of MEMS-based inertial sensors with a low-cost GPS receiver. *Sensors*, *8*(4), 2240–2261. http://dx.doi.org/10.3390/s8042240.

Sastry, K., Goldberg, D. E., & Kendall, G. (2014). Search methodologies: Introductory tutorials in optimization and decision support techniques (pp. 93–117) http://dx.doi.org/10.1007/978-1-4614-6940-7_4.

Shen, Z., Georgy, J., Korenberg, M., & Noureldin, A. (2011). Low cost two dimension navigation using an augmented Kalman filter/fast orthogonal search module for the integration of reduced inertial sensor system and Global Positioning System.

Transportation Research Part C: Emerging Technologies, *19*(6), 1111–1132. http://dx.doi.org/10.1016/j.trc.2011.01.001.

Shi, Y. (2004). *Particle swarm optimization, vol. 4*. (pp. 8–13). IEEE Neural Networks Society.

St-Pierre, M., & Gingras, D. (2004). Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system. In *IEEE intelligent vehicles symposium* (pp. 831–835). http://dx.doi.org/10.1109/IVS.2004.1336492.

Tan, X., Wang, J., Jin, S., & Meng, X. (2015). GA-SVR and pseudo-position-aided GPS/INS integration during GPS outage. *Journal of Navigation*, *68*(4), 678–696. http://dx.doi.org/10.1017/S037346331500003X.

Xu, Z., Li, Y., Rizos, C., & Xu, X. (2010). Novel hybrid of LS-SVM and Kalman filter for GPS/INS integration. *Journal of Navigation*, *63*(2), 289–299. http://dx.doi.org/10.1017/S0373463309990361.